支持事务的数据库必须具备的四大特性：

# ACID

* 原子性 全部成功/全部失败（回滚）
* 一致性 转出200，入账就是200，不是其它数字
* [隔离性](#_隔离级别：) 同步，避免并发错误
* 持久性 比如存到硬盘里

## 不隔离的问题

* 脏读 其它事务修改期间读到数据（半成品）
* 不可重复读 一次事务中多次读同一个数据期间，有其它事务UPDATE该数据

（改变目标数据：内容）

* 幻读 一次事务中多次读取数据期间，有其它事务INSERT / DELETE数据

（改变相近数据：数量）

## 隔离级别：

* Read uncommitted 读未提交：可以读到另一个事务未提交的数据（半成品）
* Read Committed 读已提交：读到的始终是最新一次commit的数据
* Repeatable Read 可重复读，一次事务期间多次查询，确保结果一致（默认不加锁，因

此本事务期间有可能被其它事务修改，导致本事务第二次查询时得不

到真实的数据（因为是本事务第一次读时创建的快照））

* Serializable 串行化，事务期间一直持有锁，该锁针对目标数据及相关数据，其它

事务无法做增、删、改操作，效率最低

从加锁的角度看隔离级别：

Read uncommitted : 不加锁

Read committed : 写操作加锁，锁行，避免多个事务同时写

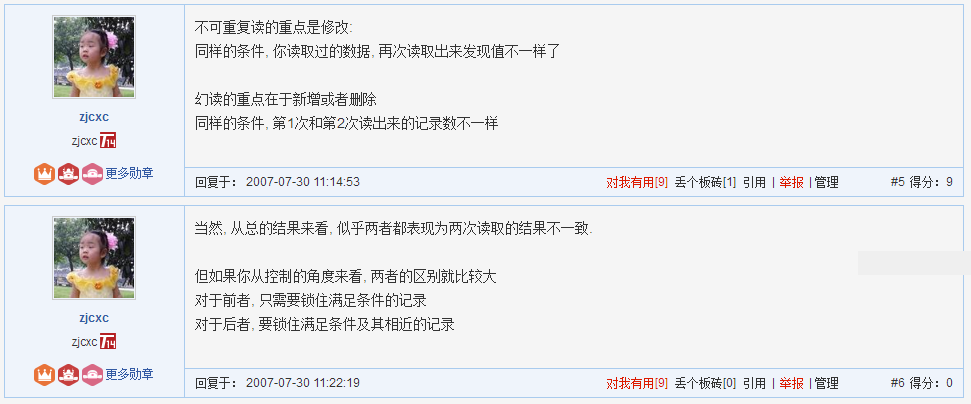
Repeatable Read : 写操作加锁，锁行。MySQL读时有快照，本事务期间多次读取数据一致

Serializable : 写操作加锁，锁表。

## 参考网址

<http://bbs.csdn.net/topics/120024254>

疑问：不可重复读和幻读感觉很像呀，区别是什么呢？

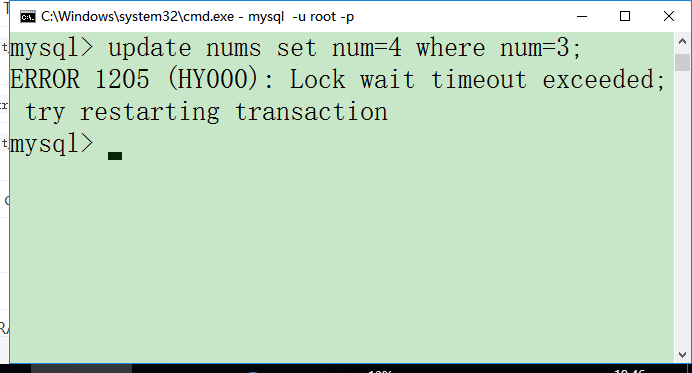


要避免不可重复读，只需要锁住满足条件的记录（防止其它事务改）

要避免幻读，要锁住满足条件的记录及相近的记录（防止其它事务增、删）

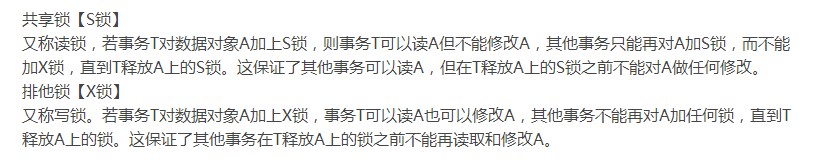
[特殊幻读](#_特殊幻读)：读到的是根本不存在的状态。

lock in share mode



<http://blog.csdn.net/xiangyubobo/article/details/52265784>

共享锁 和 排它锁



1.目的不同：S锁for读 ， X锁for写

2. X锁比S锁更严格：加了X锁,则其它事务什么锁都不能加（增删改默认加X锁，所以都不能执行，但是查询默认不加锁，可以执行。加了锁的查询操作是执行不了的。）.加了S锁,其它事务可以再加S锁。

3. X锁和S锁不能作用于同一条数据

<http://blog.csdn.net/yuwei19840916/article/details/3245107>

【MySQL共享锁和排它锁】<https://www.cnblogs.com/boblogsbo/p/5602122.html>

【界面有些血腥，内容是多方面整理的，很好】

<https://www.cnblogs.com/hadoop-dev/p/7065702.html>

1. 悲观并发控制（悲观锁）、乐观并发控制（乐观锁）。

1. 乐观锁不是锁，是通过版本号or时间戳实现并发控制的一种思想
2. 行级锁、表级锁、共享锁（S锁）、排它锁（X锁）都属于悲观锁
3. 增删改默认会加排它锁，查询默认不加锁（可以手动加S锁或者X锁）。

答疑：

Q1：MySQL为什么**可重复读**隔离级别下，其它事务可以增、删、改？

A1：本来就可以修改!!!! <http://blog.csdn.net/dong976209075/article/details/8802778>

MySQL的InnoDB引擎采用了MVCC机制（解决了幻读问题，其实只是看不到而已），

一种是快照读，不加锁（所以其它事务可以增删改）

一种是当前读，加锁，可以手动设定S锁或X锁

Q2: 加了锁的RR隔离级别和序列化级别有什么区别呢？

A2：手动加锁和自动加锁的区别。

# 附件：

## 特殊幻读

